

### **Amendments to the Claims**

Please amend the claims as shown below.

1. (Previously Presented) A correction server system, comprising:  
  
an analyzer to calculate an analytical result using at least one data entity stored in a database;  
  
a data flow manager, responsive to read requests from agents to the database, to store a read history identifying a relationship between the data entity being read and the analytical result, and  
  
a correction server that, when corrections are made to the database, identifies corrected entities in a corrected entity log and compares the corrected entity log against the read history to identify analytical results rendered possibly inconsistent due to the correction.
2. (Original) The correction server system of claim 1, further comprising the reading component, which generates a new entity from the database entity that is read and stores it in the database.
3. (Previously Presented) The correction server system of claim 1, wherein the read history log identifies leading and dependent entities, a leading entity being a database entity that is read by a component and a dependent entity being a new object entity created from the database entity that is read.
4. (Original) The correction server system of claim 1, wherein the read history stores pairs of entity identifiers.
5. (Previously Presented) The correction server system of claim 1, wherein the correction server receives correction data that includes an identifier of a database entity being corrected, an indication of fields within the database entity that are being changed and an identification of field values that are changed.
6. (Original) The correction server system of claim 5, wherein the corrected entity log stores all the correction data noted in claim 5.

7. (Original) The correction server system of claim 1, wherein the correction server further comprises a filtering agent that compares correction information to filtering criterion and stores the correction information in the corrected entity log only if the correction information matches the filtering criterion.

8. (Original) The correction server system of claim 1, wherein the correction server further includes a user interface that permits review and display of the corrected entity log, the user interface providing a "jump to" feature that, when activated with respect to an entry of the log causes a data entity referenced by the entry to be retrieved and displayed.

9. (Previously Presented) A computer-implemented correction management method, comprising:

responsive to a request to correct a first database entity, creating a second database entity that is a corrected copy of the first database entity,

storing an entry in a corrected entity log that identifies the first database entity,

comparing the corrected entity log entry against a read history log identifying prior accesses to the database,

if the entry matches an entry from the read history log, identifying a dependent database entity from the read history log as a possibly inconsistent entity, the dependent database entity based on an analytical result calculated from the first database entity.

10. (Original) The correction management method of claim 9, wherein the read history log stores paired leading entity identifiers and dependent entity identifiers relating to the prior accesses.

11. (Original) The correction management method of claim 10, wherein the comparison is made between an entity identifier from the corrected entity log and the leading entity identifier from the read history log.

12. (Original) The correction management method of claim 9, wherein the correction includes an entity identifier of the first database entity and an indication of fields within the first database entity being corrected.

13. (Original) The correction management method of claim 9, further comprising comparing the correction request to filtering criteria and performing the storing and comparing unless the correction request does not satisfy the filtering criteria.

14. (Previously Presented) Computer readable medium having stored thereon program instructions that, when executed, cause a computer system to:

responsive to a request to correct a first database entity, create a second database entity that is a corrected copy of the first database entity,

store an entry in a corrected entity log that identifies the first database entity,

compare the corrected entity log entry against a read history log identifying prior accesses to the database,

if the entry matches an entry from the read history log, identify a dependent database entity from the read history log as a possibly inconsistent entity, the dependent database entity based on an analytical result calculated from the first database entity.

15. (Original) The medium of claim 14, wherein the read history log stores paired leading entity identifiers and dependent entity identifiers relating to the prior accesses.

16. (Original) The medium of claim 15, wherein the comparison is made between an entity identifier from the corrected entity log and the leading entity identifier from the read history log.

17. (Original) The medium of claim 14, wherein the correction request includes an entity identifier of the first database entity and an indication of fields within the first database entity being corrected.

18. (Original) The medium of claim 14, further comprising comparing the correction request to filtering criteria and performing the storing and comparing unless the correction request does not satisfy the filtering criteria.

19. (Previously Presented) A system for identifying inconsistent data in a computer system, comprising:

a first database to store data generated during operation of the computer system;

a correction manager to manage corrections performed in the system, the correction manager further comprising:

a second database to store a list of corrected data entries in the first database; and

a third database to store a list of uncorrected data entries identified as potentially inconsistent due to a correction performed on an entity listed in the second database; and

a data flow manager to manage access to the first database, the second database, and the third database by an analyzer, the analyzer to provide analytical results calculated from data stored in the first database to an operator of the system.

20. (Canceled).